OIPE IAP40
OCT 1 7 2008
PATENT & TRADEMARK OFFICE

1 for AB

TRANSMITTAL OF APPEAL BRIEF (Large Entity)

Docket No. 200-0667

In Re Application Of: **Joseph G. Walacavage et al.**

Application No.	Filing Date	Examiner	Customer No.	Group Art Unit	Confirmation No.
09/966,121	September 28, 2001	K. Gebresilassie	33481	2128	4437

Invention: **METHOD OF EMULATING MACHINE TOOL BEHAVIOR FOR PROGRAMMABLE LOGIC CONTROLLER LOGICAL VERIFICATION SYSTEM**

COMMISSIONER FOR PATENTS:

Transmitted herewith is the Appeal Brief in this application, with respect to the Notice of Appeal filed on:
August 14, 2008

The fee for filing this Appeal Brief is: **$540.00**

☐ A check in the amount of the fee is enclosed.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 06-1510. I have enclosed a duplicate copy of this sheet.

☐ Payment by credit card. Form PTO-2038 is attached.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Signature

Dated: **October 14, 2008**

Daniel H. Bliss (Reg. No. 32,398) [0693.00258]
Bliss McGlynn, P.C.
2075 West Big Beaver Road, Suite 600
Troy, Michigan 48084
(248) 649-6090

Record I.D. 81047711

cc:

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on
October 14, 2008
(Date)

Signature of Person Mailing Correspondence

Daniel H. Bliss
Typed or Printed Name of Person Mailing Correspondence

OCT 1 7 2008
OIPE IAP40
PATENT & TRADEMARK OFFICE

TRANSMITTAL OF APPEAL BRIEF (Large Entity)	Docket No. 200-0667

In Re Application Of: **Joseph G. Walacavage et al.**

Application No.	Filing Date	Examiner	Customer No.	Group Art Unit	Confirmation No.
09/966,121	September 28, 2001	K. Gebresilassie	33481	2128	4437

Invention: **METHOD OF EMULATING MACHINE TOOL BEHAVIOR FOR PROGRAMMABLE LOGIC CONTROLLER LOGICAL VERIFICATION SYSTEM**

COMMISSIONER FOR PATENTS:

Transmitted herewith is the Appeal Brief in this application, with respect to the Notice of Appeal filed on: **August 14, 2008**

The fee for filing this Appeal Brief is: **$540.00**

☐ A check in the amount of the fee is enclosed.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 06-1510. I have enclosed a duplicate copy of this sheet.

☐ Payment by credit card. Form PTO-2038 is attached.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Signature

Dated: **October 14, 2008**

Daniel H. Bliss (Reg. No. 32,398) [0693.00258]
Bliss McGlynn, P.C.
2075 West Big Beaver Road, Suite 600
Troy, Michigan 48084
(248) 649-6090

Record I.D. 81047711

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on

October 14, 2008
(Date)

Signature of Person Mailing Correspondence

Daniel H. Bliss
Typed or Printed Name of Person Mailing Correspondence

cc:

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Art Unit:	2128	
Examiner:	K. Gebresilassie	
Applicant(s):	J. G. Walacavage et al.	
Serial No.:	09/966,121	**APPEAL BRIEF**
Filing Date:	September 28, 2001	
For:	METHOD OF EMULATING MACHINE TOOL BEHAVIOR FOR PROGRAMMABLE LOGIC CONTROLLER LOGICAL VERIFICATION SYSTEM	

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

By Notice of Appeal filed August 14, 2008, Applicants have appealed the Final Rejection dated May 14, 2008 and submit this brief in support of that appeal.

REAL PARTY IN INTEREST

The real party in interest is the Assignee, Ford Global Technologies, Inc.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences regarding the present application.

CERTIFICATE OF MAILING: (37 C.F.R. 1.8) I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the U.S. Postal Service with sufficient postage as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450 on October 14, 2008, by ______.
Daniel H. Bliss

STATUS OF CLAIMS

Claims 1 through 19 have been rejected.

Claims 1 through 19 are being appealed.

STATUS OF AMENDMENTS

An Amendment Under 37 C.F.R. 1.116 was filed on July 14, 2008 in response to the Final Office Action dated May 14, 2008. An Advisory Action dated August 5, 2008 was issued and did not indicate that the Amendment under 37 C.F.R. 1.116 would be entered. A Notice of Appeal, along with the requisite fee, was filed on August 14, 2008. The Appeal Brief, along with the requisite fee, is submitted herewith.

SUMMARY OF THE CLAIMED SUBJECT MATTER

Independent Claim 1

The claimed subject matter of independent claim 1 is directed to a method of using transformational arrays to emulate model behavior for a programmable logic controller logical verification system. [Referring to FIG. 2, a method, according to the present invention, of emulating machine tool behavior as part of the PLC logical verification system 18 is shown. In general, the method includes generating transformational arrays based on CAD geometries during the design phase of the machinery for the mechanical model. The transformational arrays are movies of manipulation of individual components in the mechanical model and are generated with the mechanical tool design system 16. These transformational arrays are then associated with the particular piece of machine component, such as a clamp, throughout the life cycle of the

design and verification process. Once the transformational arrays exist and the mechanical design is implemented on a CAD system to produce the mechanical model, the method includes exporting the mechanical model to a special purpose viewer or motion player 30 such as VisLine. Within the motion player 30, these transformational arrays are sequenced to give a first pass rendition of what the overall machine or manufacturing line behavior will be. After the user 12 is satisfied that the behavior is acceptable or the one desired, then all this information is exported to the controls system design system 17. The controls system design system 17 converts the sequence of the transformational arrays to PLC code. The user 12, using the controls system design system 17, has now substituted the sequencing logic of the transformational arrays that was present in the first pass rendition with the PLC code that is intended to be executed on the plant floor. The user 12 exports the PLC code to the PLC emulator 20 to play and visualize the PLC code. By playing this PLC code in the PLC emulator 20, and through the binding that has been preserved through this process, the user 12 can observe the sequencing of these transformational arrays using the actual PLC code as if they were watching a machine or manufacturing line of a vehicle assembly plant floor.] (FIG. 2; Specification, page 9, line 6 through page 11, line 6).

The method includes the steps of constructing a mechanical model using a computer. [The method also includes using the mechanical tool design system to construct a mechanical model 26. The mechanical model 26 is a portion of a simulated assembly or manufacturing line representing actual machines, as opposed to control hardware. The mechanical model 26 contains CAD geometries and mechanical motion from the electro-mechanical model. The geometry representation is defined through the use of (NURB) type equations. In the present invention, a user 12 uses a computer 14 for carrying out the emulation.

The computer 14 sends and receives information from a mechanical tool design system 16 via an electronic link. The mechanical tool design system 16 is a computer aided design (CAD) system and provides engineering data for standard components, tools, fixture models, and robots to create a mechanical model. The mechanical tool design system 16 allows the user 12 to define a set of kinematics or motions for the geometry of individual components in the mechanical model.] (FIGS. 1 and 2; Specification, page 7, lines 14 through 22 and page 11, line 16 through page 12, line 2).

The method also includes the steps of generating transformational arrays for the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer. [The method includes generating transformational arrays 28 for the mechanical model 26 with the mechanical tool design system 16. The transformational arrays 28 are a set of coordinates in each of a plurality of arrays. The set of coordinates comprise six coordinates of three dimensions and three rotations. The transformational arrays 28 translate or transform the original coordinates for each point on the CAD geometry into a linear file of recorded coordinates. The transformational arrays are generated by incrementally recording one position of a specific piece of geometry as it is moved through space over a period of time. It should be appreciated that the user 12 of the CAD software for the mechanical tool design system 16 is the creator of the path of the mechanical element in the mechanical model.] (FIGS. 1 and 2; Specification, page 12, lines 3 through 16).

The method further includes the steps of viewing motion of the mechanical model in a motion viewer based on the transformational arrays using the computer. [The method includes constructing a motion file based on the mechanical model 26 and the transformational

arrays 28 and viewing motion of the mechanical model 26 based on the transformational arrays 28 using a motion player 30 to play the motion file. A motion file is a series of transformations that represent the motion path of a three-dimensional object through simulation space. The motion player 30 interfaces with the mechanical model 26 to control when and in what direction a given motion file plays. The motion player 30 is a software product known as VisLine that provides a lightweight visual animation capability based on CAD three-dimensional models and sets of transformations called "motion files" derived from the CAD models. The order and time for the playing of a motion file is determined by executing a program written in the PLC emulator 20. The written program is a language specification supported by the PLC emulator 20, which is capable of communicating a complex sequence for the execution of motion files and is based on the execution of specified events according to specified conditions. It should be appreciated that events cause actions to occur and create signals available as conditions for executing other events. It should also be appreciated that the transformation does not change nature of object or part but moves the object or part in space.] (FIGS. 1 and 2; Specification, page 12, line 17 through page 13, line 18).

The method includes the steps of determining whether the motion of the mechanical model is acceptable. [The method further includes determining whether the motion of the mechanical model 26 is acceptable. The user 12 views the motion or behavior of the mechanical model 26 and determines whether the behavior is acceptable or desired.] (FIGS. 1 and 2; Specification, page 13, lines 19 through 22).

The method includes the steps of replicating the motion of the mechanical model by generating a PLC code for the motion of the mechanical model using the computer if the motion of the mechanical model was acceptable. [If the behavior is acceptable, the method

exports all of this information to the controls system design system 17, which converts this information to PLC code to build a manufacturing line. The user 12 replicates the motion of the mechanical model 26 previously defined in the mechanical model 26 with motion commands in PLC code using the controls system design system 17. If the user 12 determines that the behavior is not acceptable or desired, the method includes returning to the mechanical tool design system 16 to change the mechanical model 26 as illustrated in FIG. 3.] (FIGS. 1 through 3; Specification, page 13, line 22 through page 14, line 10).

The method includes the steps of using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator. [After the sequencing logic of the transformational arrays has been replicated with PLC code, the user 12 exports the PLC code to the PLC emulator 20 to play and visualize the PLC code. The user 12 compares the behavior of the PLC code to the accepted motion of the mechanical model 26 through the sequencing of the transformational arrays 28 as if they were watching a machine or manufacturing line of a vehicle assembly plant floor. It should be appreciated that the method is carried out on the computer 14 by the user 12.] (FIGS. 1 and 2; Specification, page 14, lines 11 through 14).

Independent claim 11

The claimed subject matter of independent claim 11 is directed to a method of using transformational arrays to emulate model behavior for a programmable logic controller logical verification system. [Referring to FIG. 2, a method, according to the present invention, of emulating machine tool behavior as part of the PLC logical verification system 18 is shown. In general, the method includes generating transformational arrays based on CAD geometries during

the design phase of the machinery for the mechanical model. The transformational arrays are movies of manipulation of individual components in the mechanical model and are generated with the mechanical tool design system 16. These transformational arrays are then associated with the particular piece of machine component, such as a clamp, throughout the life cycle of the design and verification process. Once the transformational arrays exist and the mechanical design is implemented on a CAD system to produce the mechanical model, the method includes exporting the mechanical model to a special purpose viewer or motion player 30 such as VisLine. Within the motion player 30, these transformational arrays are sequenced to give a first pass rendition of what the overall machine or manufacturing line behavior will be. After the user 12 is satisfied that the behavior is acceptable or the one desired, then all this information is exported to the controls system design system 17. The controls system design system 17 converts the sequence of the transformational arrays to PLC code. The user 12, using the controls system design system 17, has now substituted the sequencing logic of the transformational arrays that was present in the first pass rendition with the PLC code that is intended to be executed on the plant floor. The user 12 exports the PLC code to the PLC emulator 20 to play and visualize the PLC code. By playing this PLC code in the PLC emulator 20, and through the binding that has been preserved through this process, the user 12 can observe the sequencing of these transformational arrays using the actual PLC code as if they were watching a machine or manufacturing line of a vehicle assembly plant floor.] (FIG. 2; Specification, page 9, line 6 through page 11, line 6).

The method includes the steps of constructing a mechanical model using a computer. [The method also includes using the mechanical tool design system to construct a mechanical model 26. The mechanical model 26 is a portion of a simulated assembly or

manufacturing line representing actual machines, as opposed to control hardware. The mechanical model 26 contains CAD geometries and mechanical motion from the electro-mechanical model. The geometry representation is defined through the use of (NURB) type equations. In the present invention, a user 12 uses a computer 14 for carrying out the emulation. The computer 14 sends and receives information from a mechanical tool design system 16 via an electronic link. The mechanical tool design system 16 is a computer aided design (CAD) system and provides engineering data for standard components, tools, fixture models, and robots to create a mechanical model. The mechanical tool design system 16 allows the user 12 to define a set of kinematics or motions for the geometry of individual components in the mechanical model.] (FIGS. 1 and 2; Specification, page 7, lines 14 through 22 and page 11, line 16 through page 12, line 2).

The method also includes the steps of generating CAD transformational arrays for motion in the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer. [The method includes generating transformational arrays 28 for the mechanical model 26 with the mechanical tool design system 16. The transformational arrays 28 are a set of coordinates in each of a plurality of arrays. The set of coordinates comprise six coordinates of three dimensions and three rotations. The transformational arrays 28 translate or transform the original coordinates for each point on the CAD geometry into a linear file of recorded coordinates. The transformational arrays are generated by incrementally recording one position of a specific piece of geometry as it is moved through space over a period of time. It should be appreciated that the user 12 of the CAD software for the mechanical tool design system 16 is the

creator of the path of the mechanical element in the mechanical model.] (FIG. 1; Specification, page 12, lines 3 through 16).

The method includes the steps of constructing a motion file based on the mechanical model and the CAD transformational arrays using the computer and viewing the motion of the motion file in a motion viewer using the computer. [The method includes constructing a motion file based on the mechanical model 26 and the transformational arrays 28 and viewing motion of the mechanical model 26 based on the transformational arrays 28 using a motion player 30 to play the motion file. A motion file is a series of transformations that represent the motion path of a three-dimensional object through simulation space. The motion player 30 interfaces with the mechanical model 26 to control when and in what direction a given motion file plays. The motion player 30 is a software product known as VisLine that provides a lightweight visual animation capability based on CAD three-dimensional models and sets of transformations called "motion files" derived from the CAD models. The order and time for the playing of a motion file is determined by executing a program written in the PLC emulator 20. The written program is a language specification supported by the PLC emulator 20, which is capable of communicating a complex sequence for the execution of motion files and is based on the execution of specified events according to specified conditions. It should be appreciated that events cause actions to occur and create signals available as conditions for executing other events. It should also be appreciated that the transformation does not change nature of object or part but moves the object or part in space.] (FIGS. 1 and 2; Specification, page 12, line 17 through page 13, line 18).

The method includes the steps of determining whether the motion of the mechanical model is acceptable. [The method further includes determining whether the motion

of the mechanical model 26 is acceptable. The user 12 views the motion or behavior of the mechanical model 26 and determines whether the behavior is acceptable or desired.] (FIGS. 1 and 2; Specification, page 13, lines 19 through 22).

The method includes the steps of replicating the motion of the mechanical model with motion commands in a PLC code using the computer if the motion of the mechanical model was acceptable. [If the behavior is acceptable, the method exports all of this information to the controls system design system 17, which converts this information to PLC code to build a manufacturing line. The user 12 replicates the motion of the mechanical model 26 previously defined in the mechanical model 26 with motion commands in PLC code using the controls system design system 17. If the user 12 determines that the behavior is not acceptable or desired, the method includes returning to the mechanical tool design system 16 to change the mechanical model 26 as illustrated in FIG. 3.] (FIGS. 1 through 3; Specification, page 13, line 22 through page 14, line 10).

The method includes the steps of using the accepted motion of the mechanical model to compare the behavior of the PLC code to the accepted motion by playing the PLC code with a PLC emulator. [After the sequencing logic of the transformational arrays has been replicated with PLC code, the user 12 exports the PLC code to the PLC emulator 20 to play and visualize the PLC code. The user 12 compares the behavior of the PLC code to the accepted motion of the mechanical model 26 through the sequencing of the transformational arrays 28 as if they were watching a machine or manufacturing line of a vehicle assembly plant floor. It should be appreciated that the method is carried out on the computer 14 by the user 12.] (FIGS. 1 and 2; Specification, page 14, lines 11 through 14).

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

The ground of rejection to be reviewed on appeal is whether the claimed invention of claims 1 through 19 is disclosed and anticipated under 35 U.S.C. § 102(e) by Walacavage et al. (U.S. Patent No. 6,442,441).

ARGUMENT

Claims Not Disclosed or Anticipated Under 35 U.S.C. § 102(e)

As to patentability, 35 U.S.C. § 102(e) provides that a person shall be entitled to a patent unless:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for a patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent . . .

A rejection grounded on anticipation under 35 U.S.C. § 102 is proper only where the subject matter claimed is identically disclosed or described in a reference. In other words, anticipation requires the presence of a single prior art reference which discloses each and every element of the claimed invention arranged as in the claim. <u>In re Arkley</u>, 455 F.2d 586, 172 U.S.P.Q. 524 (C.C.P.A. 1972); <u>Kalman v. Kimberly-Clark Corp.</u>, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983); <u>Lindemann Maschinenfabrik GMBH v. American Hoist & Derrick Co.</u>, 730 F.2d 1452, 221 U.S.P.Q. 481 (Fed. Cir. 1984).

As to the reference applied by the Examiner, U.S. Patent No. 6,442,441 to Walacavage discloses a method of automatically generating and verifying programmable logic controller code. The method includes the steps of constructing a neutral control model file,

determining whether the neutral control model file is correct and generating programmable logic controller (PLC) code if the neutral control model file is correct. The method also includes the steps of verifying whether the PLC code is correct and using the PLC code by a PLC to build a tool if the PLC code is correct. Walacavage does not disclose generating transformational arrays for a mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using a computer and viewing motion of the mechanical model in a motion viewer based on the transformation arrays using the computer to determine whether the motion of the mechanical model is acceptable. Walacavage also does not disclose replicating a motion of a mechanical model by generating a PLC code for the motion of the mechanical model using a computer if the motion of the mechanical model was acceptable and using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator.

Claims 1 through 10

In contradistinction, claim 1 claims the present invention as a method of using transformational arrays to emulate model behavior for a programmable logic controller logical verification system. The method includes the steps of constructing a mechanical model using a computer, generating transformational arrays for the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer, viewing motion of the mechanical model in a motion viewer based on the transformation arrays using the computer, and determining whether the motion of the mechanical model is acceptable. The method also includes the steps of replicating the motion of the mechanical model by generating a PLC code for the motion of the mechanical

model using the computer if the motion of the mechanical model was acceptable and using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator.

As to the differences between the prior art and the claims at issue, Walacavage '441 merely discloses a method of automatically generating and verifying programmable logic controller code by generating programmable logic controller (PLC) code if a neutral control model file is correct, verifying whether the PLC code is correct, and using the PLC code by a PLC to build a tool if the PLC code is correct. Walacavage '441 lacks the steps of generating transformational arrays for a mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using a computer and viewing motion of the mechanical model in a motion viewer based on the transformation arrays using the computer to determine whether the motion of the mechanical model is acceptable. In Walacavage '441, the method uses a neutral control model file, converts the neutral control model file into a compatible PLC code and analytically verifies the PLC code. In Walacavage '441, it is clear in Column 2, line 54, through Column 3, lines 1 through 40, that the method uses a neutral control model file and not transformational arrays to incrementally record one position of each piece of geometry of the mechanical model moved through space over a period of time.

Contrary to the Examiner's opinion, the term "transformational arrays" is not merely another name for a "neutral control model file", but is a term consistent with that disclosed in the specification. The claim language is to be read in view of the specification as it would be interpreted by one of ordinary skill in the art. In re Morris, 127 F.3d 1048, 1053-54, 44 U.S.P.Q.2d 1023, 1027 (Fed. Cir. 1997). In the present application, on page 9, lines 9 through

14, the method includes generating transformational arrays based on CAD geometries during the design phase of the machinery for the mechanical model. The transformational arrays are movies of manipulation of individual components in the mechanical model and are generated with the mechanical tool design system 16 and not a neutral control model file. As a result, the neutral control model file is not the same thing as a mechanical model with the transformational arrays of the present application. As stated on page 10, lines 3 through 6 of the present application, within the motion player 30, these transformational arrays are sequenced to give a first pass rendition of what the overall machine or manufacturing line behavior will be, which is entirely different from the neutral control model file of Walacavage '441. Therefore, Walacavage '441 does not perform the steps of generating transformational arrays for a mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using a computer and viewing motion of the mechanical model in a motion viewer based on the transformation arrays using the computer to determine whether the motion of the mechanical model is acceptable as claimed by Applicants. As such, the claim language cannot be interpreted to read on the neutral control model file of Walacavage '441.

Walacavage '441 also lacks replicating a motion of a mechanical model by generating a PLC code for the motion of the mechanical model using a computer if the motion of the mechanical model was acceptable and using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator. In Walacavage '441, there is a special purpose viewer or motion player such as VisLine in block 40, but there is no additional PLC emulator to play the PLC code such that the user can observe the motion of the mechanical model using the actual PLC code as if they were watching a machine or manufacturing line of a vehicle assembly plant floor. In

Walacavage '441, there is a virtual PLC generator 15 that generates PLC code and a line verification system 14 that verifies the PLC code. In Walacavage '441, it is clear in Column 2, lines 37 and 39, that the line verification system 14 verifies the PLC code for the line model and not the PLC generator 15.

Contrary to the Examiner's opinion, the term "emulator" is not merely another name for a "PLC code generator", but is a term consistent with that disclosed in the specification. The claim language is to be read in view of the specification as it would be interpreted by one of ordinary skill in the art. In re Morris, 127 F.3d 1048, 1053-54, 44 U.S.P.Q.2d 1023, 1027 (Fed. Cir. 1997). In the present application, on page 8, lines 18 and 19, although the emulator 20 sends and receives signals with the PLC logical verification system 18, it is the verification system 18 that verifies the PLC code and not the emulator 20. As a result, the PLC generator 15 is not the same thing as the emulator 20 of the present application. As stated on page 14, lines 11 through 14 of the present application, the user 12 exports the PLC code to the PLC emulator 20 to play and visualize the PLC code, which is entirely different from the PLC code generator 15 of Walacavage '441. Therefore, Walacavage '441 does not perform the steps of replicating a motion of a mechanical model by generating a PLC code for the motion of the mechanical model using a computer if the motion of the mechanical model was acceptable and using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator as claimed by Applicants. As such, the claim language cannot be interpreted to read on the PLC generator 15 of Walacavage '441.

Based on the above, Walacavage '441 fails to disclose the combination of a method of using transformational arrays to emulate model behavior for a programmable logic controller logical verification system including the steps of constructing a mechanical model

using a computer, generating transformational arrays for a mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer, viewing motion of the mechanical model in a motion viewer based on the transformational arrays using the computer, determining whether the motion of the mechanical model is acceptable, replicating the motion of the mechanical model by generating a PLC code for the motion of the mechanical model using the computer if the motion of the mechanical model was acceptable, and using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator as claimed by Applicants.

Against this background, it is submitted that the present invention of claims 1 through 10 is not anticipated in view of Walacavage '441. The reference cannot be used to anticipate the method of using transformational arrays to emulate model behavior for a programmable logic controller logical verification system of claims 1 through 10. Therefore, it is respectfully submitted that claims 1 through 10 are not anticipated and are allowable over the rejection under 35 U.S.C. § 102(e).

Claims 11 through 19

As to independent claim 11, claim 11 claims the present invention as a method of using transformational arrays to emulate model behavior for a programmable logic controller logical verification system. The method includes the steps of constructing a mechanical model using a computer and generating CAD transformational arrays for motion in the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer. The method also includes

the steps of constructing a motion file based on the mechanical model and the CAD transformational arrays using the computer and viewing the motion of the motion file in a motion viewer using the computer. The method includes the steps of determining whether the motion of the mechanical model is acceptable and replicating the motion of the mechanical model with motion commands in a PLC code using the computer if the motion of the mechanical model was acceptable. The method further includes the steps of using the accepted motion of the mechanical model to compare the behavior of the PLC code to the accepted motion by playing the PLC code with a PLC emulator.

As to the differences between the prior art and the claims at issue, Walacavage '441 <u>merely</u> discloses a method of automatically generating and verifying programmable logic controller code by generating programmable logic controller (PLC) code if a neutral control model file is correct, verifying whether the PLC code is correct, and using the PLC code by a PLC to build a tool if the PLC code is correct. Walacavage '441 lacks the steps of generating CAD transformational arrays for motion in a mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using a computer and viewing the motion of the motion file in a motion viewer using the computer to determine whether the motion of the mechanical model is acceptable. In Walacavage '441, the method uses a neutral control model file, converts the neutral control model file into a compatible PLC code and analytically verifies the PLC code. In Walacavage '441, it is clear in Column 2, line 54, through Column 3, lines 1 through 40, that the method uses a neutral control model file and <u>not</u> CAD transformational arrays for motion in a mechanical model to incrementally record one position of each piece of geometry of the mechanical model moved through space over a period of time.

Contrary to the Examiner's opinion, the term "CAD transformational arrays" is not merely another name for a "neutral control model file", but is a term consistent with that disclosed in the specification. The claim language is to be read in view of the specification as it would be interpreted by one of ordinary skill in the art. In re Morris, 127 F.3d 1048, 1053-54, 44 U.S.P.Q.2d 1023, 1027 (Fed. Cir. 1997). In the present application, on page 9, lines 9 through 14, the method includes generating transformational arrays based on CAD geometries during the design phase of the machinery for the mechanical model. The transformational arrays are movies of manipulation of individual components in the mechanical model and are generated with the mechanical tool design system 16 and not a neutral control model file. As a result, the neutral control model file is not the same thing as a mechanical model with the CAD transformational arrays of the present application. As stated on page 10, lines 3 through 6 of the present application, within the motion player 30, these CAD transformational arrays are sequenced to give a first pass rendition of what the overall machine or manufacturing line behavior will be, which is entirely different from the neutral control model file of Walacavage '441. Therefore, Walacavage '441 does not perform the steps of generating CAD transformational arrays for motion in a mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using a computer and viewing the motion of the motion file in a motion viewer using the computer to determine whether the motion of the mechanical model is acceptable as claimed by Applicants. As such, the claim language cannot be interpreted to read on the neutral control model file of Walacavage '441.

Walacavage '441 also lacks replicating a motion of a mechanical model with motion commands in a PLC code using a computer if the motion of the mechanical model was

acceptable and using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator. In Walacavage '441, there is a special purpose viewer or motion player (such as VisLine in block 40), but there is no additional PLC emulator to play the PLC code such that the user can observe the motion of the mechanical model using the actual PLC code as if they were watching a machine or manufacturing line of a vehicle assembly plant floor. In Walacavage '441, there is a virtual PLC generator 15 that generates PLC code and a line verification system 14 that verifies the PLC code. In Walacavage '441, it is clear in Column 2, lines 37 and 39, that the line verification system 14 verifies the PLC code for the line model and not the PLC generator 15.

Contrary to the Examiner's opinion, the term "emulator" is not merely another name for a "PLC code generator", but is a term consistent with that disclosed in the specification. The claim language is to be read in view of the specification as it would be interpreted by one of ordinary skill in the art. In re Morris, 127 F.3d 1048, 1053-54, 44 U.S.P.Q.2d 1023, 1027 (Fed. Cir. 1997). In the present application, on page 8, lines 18 and 19, although the emulator 20 sends and receives signals with the PLC logical verification system 18, it is the verification system 18 that verifies the PLC code and not the emulator 20. As a result, the PLC generator 15 is not the same thing as the emulator 20 of the present application. As stated on page 14, lines 11 through 14 of the present application, the user 12 exports the PLC code to the PLC emulator 20 to play and visualize the PLC code, which is entirely different from the PLC code generator 15 of Walacavage '441. Therefore, Walacavage '441 does not perform the steps of replicating a motion of a mechanical model with motion commands in a PLC code using a computer if the motion of the mechanical model was acceptable and using the accepted motion of the mechanical model to compare the behavior of the PLC code to the accepted motion by playing the PLC code

with a PLC emulator as claimed by Applicants. As such, the claim language cannot be interpreted to read on the PLC generator 15 of Walacavage '441.

Based on the above, Walacavage '441 fails to disclose the combination of a method of using transformational arrays to emulate model behavior for a programmable logic controller logical verification system including the steps of constructing a mechanical model using a computer, generating CAD transformational arrays for motion in the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer, constructing a motion file based on the mechanical model and the CAD transformational arrays using the computer, viewing the motion of the motion file in a motion viewer using the computer, determining whether the motion of the mechanical model is acceptable, replicating the motion of the mechanical model with motion commands in a PLC code using the computer if the motion of the mechanical model was acceptable, and using the accepted motion of the mechanical model to compare the behavior of the PLC code to the accepted motion by playing the PLC code with a PLC emulator as claimed by Applicants.

Against this background, it is submitted that the present invention of claims 11 through 19 is not anticipated in view of Walacavage '441. The reference cannot be applied to anticipate the combination of the method of using transformational arrays to emulate model behavior for a programmable logic controller logical verification system of claims 11 through 19. Therefore, it is respectfully submitted that claims 11 through 19 are not anticipated and are allowable over the rejection under 35 U.S.C. § 102(e).

CONCLUSION

In conclusion, it is respectfully submitted that the rejection of claims 1 through 19 is improper and should be reversed.

Respectfully submitted,

By: Daniel H. Bliss
Registration No. 32,398

BLISS McGLYNN, P.C.
2075 West Big Beaver Road
Suite 600
Troy, Michigan 48084-3443
(248) 649-6090

Dated: October 14, 2008

Attorney Docket No.: 0693.00258
Ford Disclosure No.: 200-0667

CLAIMS APPENDIX

The claims on appeal are as follows:

1. A method of using transformational arrays to emulate model behavior for a programmable logic controller logical verification system, said method comprising the steps of:

constructing a mechanical model using a computer;

generating transformational arrays for the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer;

viewing motion of the mechanical model in a motion viewer based on the transformational arrays using the computer;

determining whether the motion of the mechanical model is acceptable;

replicating the motion of the mechanical model by generating a PLC code for the motion of the mechanical model using the computer if the motion of the mechanical model was acceptable; and

using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator.

2. A method as set forth in claim 1 wherein said step of constructing comprises using a mechanical tool design system to construct the mechanical model.

3. A method as set forth in claim 2 including the step of constructing an electromechanical model.

4. A method as set forth in claim 3 wherein said step of constructing the mechanical model includes binding the electromechanical model to the mechanical model.

5. A method as set forth in claim 4 wherein said step of constructing the electromechanical model comprises using a PLC logical verification system to construct the electromechanical model.

6. A method as set forth in claim 1 wherein said step of generating transformational arrays comprises generating the transformational arrays based on computer aided design (CAD) geometries of the mechanical model.

7. A method as set forth in claim 6 including the step of exporting the mechanical model to a control system design system.

8. A method as set forth in claim 7 including the step of constructing a motion file based on the mechanical model and transformational arrays.

9. A method as set forth in claim 8 wherein said step of displaying further comprises playing the motion file by a motion player.

10. A method as set forth in claim 8 including the step of returning to the mechanical tool design system if the motion of the mechanical model is not acceptable.

11. A method of using transformational arrays to emulate model behavior for a programmable logic controller logical verification system, said method comprising the steps of:

constructing a mechanical model using a computer;

generating CAD transformational arrays for motion in the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer;

constructing a motion file based on the mechanical model and the CAD transformational arrays using the computer;

viewing the motion of the motion file in a motion viewer using the computer;

determining whether the motion of the mechanical model is acceptable;

replicating the motion of the mechanical model with motion commands in a PLC code using the computer if the motion of the mechanical model was acceptable; and

using the accepted motion of the mechanical model to compare the behavior of the PLC code to the accepted motion by playing the PLC code with a PLC emulator.

12. A method as set forth in claim 11 wherein said step of constructing comprises using a mechanical tool design system to construct the mechanical model.

13. A method as set forth in claim 12 including the step of constructing an electromechanical model.

14. A method as set forth in claim 13 wherein said step of constructing the mechanical model includes binding the electromechanical model to the mechanical model.

15. A method as set forth in claim 14 wherein said step of constructing the electromechanical model comprises using a control system design system to construct the electromechanical model.

16. A method as set forth in claim 11 wherein said step of generating comprises generating CAD transformational arrays based on computer aided design (CAD) geometries of the mechanical model.

17. A method as set forth in claim 11 including the step of exporting the mechanical model to the PLC emulator.

18. A method as set forth in claim 11 wherein said step of displaying further comprises playing the motion file by a motion player.

19. A method as set forth in claim 11 including the step of returning to the mechanical tool design system if the motion of the mechanical model is not acceptable.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None